

Message Hiding Using the Least Significant Bit Method with Shifting Hill Cipher Security

Syafrie Naufal Mahendra¹, Fikri Budiman*²

^{1,2}Faculty of Computer Science, Dian Nuswantoro University

Imam Bonjol Street no.207 Semarang

E-mail : fikri.budiman@dsn.dinus.ac.id

*Corresponding author

Abstract - Technological developments go hand in hand with advances in digital messaging. In protecting the confidentiality of the message, it is necessary to double secure the data. This security can be done with a combination of steganography and cryptographic techniques. Steganography algorithm which is a technique for hiding messages well, one of which is Least Significant Bit (LSB). The LSB algorithm is a simple method because it only converts the value of the last bit in a message with the inserted message bit, which is a convenience of the LSB algorithm, but it becomes vulnerable to message theft attacks if not combined with other algorithms for security. So it is necessary to increase security. This research developed a combination method of LSB algorithm for steganography technique with Hill Cipher algorithm for cryptographic technique, Hill Cipher was developed with shifting (shifting) 2 (two) characters. With the development of this method, hackers will find it difficult to crack messages, and is expected to improve the performance of the algorithm in affecting image quality and travel time in running the algorithm. The results of this study will be tested using several evaluation tools MSE, PSNR, BER, CER, AE, and Entropy. With the development of this method, hackers will find it difficult to decipher messages, and from the results of this experiment has been able to improve the performance of the algorithm in maintaining image quality and can shorten travel time in running the algorithm.

Keywords - Least Significant Bit (LSB), Hill Cipher, Shifting, Cryptography, Steganography

1. INTRODUCTION

Documents In this digital era, sending messages is important, especially in terms of maintaining the confidentiality of the message. One of the commonly used data security techniques is to insert or insert a message into a media and make it into a random message that cannot be recognized. This security technique is called cryptography and steganography.

Cryptography is the technique of converting the original message using a key into a message that cannot be recognized by someone who does not have the key [1]. Meanwhile, steganography is a method of hiding messages into a medium so that the message can no longer be recognized [1, 2].

Cryptography is currently categorized into two lifetimes, namely classical cryptography and modern cryptography [3-5]. Classical cryptography has the basic principle of shifting characters so that later the characters are composed into unreadable words and sentences. Modern cryptography, on the other hand, converts characters into binary numbers which are then processed and arranged into words and sentences that cannot be read as well. Based on

the key, cryptography is divided into two categories, namely cryptography with symmetric keys and cryptography with asymmetric keys [3, 5-7]. The difference between symmetric key cryptography and asymmetric keys lies in the keys for encryption and decryption [5]. When viewed from the differences, cryptography has only one purpose, which is to hide the original message so that it cannot be seen by others. However, cryptography alone is not enough to protect the message, there needs to be a combination. One method for combining cryptography is steganography.

Symmetric cryptography techniques are faster at running algorithms than asymmetric cryptography [2, 7]. This is because symmetric cryptography uses permutations and substitution calculations [7, 8]. One form of symmetric cryptography is Hill Cipher. Hill Cipher has several advantages in data encryption. Hill ciphers have a matrix key [1, 2], and are used to encrypt messages by multiplying the value of a character by that key. Hill Cipher is a very simple and fast method of processing [8-10].

Previous research explains that steganography has a flow in the same direction as the cryptographic flow [11], because, steganography has the purpose of hiding secret messages through a medium. The steganography algorithm that is usually used is Least Significant Bit (LSB). LSB is a simple method because it only changes the value of the last bit in a message with the inserted message bit [1, 12-14]. However, from the simplicity and ease of the LSB algorithm, this method is vulnerable to attack if not combined with other algorithms.

Having tested a combination of Hill Cipher algorithm to encrypt text messages and LSB algorithm to hide text messages into image media [15, 16], the trial proved that by combining Hill Cipher and LSB algorithms can increase security without damaging image media. However, these trials need to be carried out more in-depth development by increasing the algorithm from two levels to three levels. Because previous studies only used the ordinary Hill Cipher algorithm.

One of the commonly used data security techniques is to insert or insert a message into a media and make it into a random message that cannot be recognized. The security technique of symmetric cryptography is a hill chipper. Hill Cipher and LSB can increase security without damaging the image media. To improve its performance, it is necessary to conduct experiments in shifting letters which to the author's knowledge has not been done in previous researches. This needs to be done experiments to determine the effect on image quality and travel time in running the algorithm.

In this case, the author will improve the Hill Cipher algorithm by shifting letters as many as two characters forward. Shifting letters is used to confuse crypto analysis and avoid known plaintext-ciphertext attacks [10, 17]. So it is expected that the results of Hill Cipher algorithm encryption with improvements will be different from ordinary Hill Cipher algorithms. This research is expected to improve the performance of the algorithm in affecting image quality and travel time in running the algorithm, namely the development of a method by randomizing a text message into an unrecognizable message using the Hill Cipher algorithm by shifting letters as much as two characters forward and inserting the message into an RGB image object using the LSB algorithm and not deform the object of the picture.

The proposed method will be evaluated for success by measuring the values of Mean Square Error (MSE), Peak Signal to Noise Ratio (PSNR), Bit Error Rate (BER), Avalanche Effect (AE), Entropy (e) for digital image change, and Bit Error Rate (BER), Character Error Rate (CER) for text change.

2. RESEARCH METHOD

The dataset used in this experiment consisted of text from the alphabet letters from 'a' to 'z', and RGB color images obtained from The USC-sipi Image Database. The length of the text used is 32, 64, 128, and 14,628 characters. The use of such character lengths is used to prove the insertion of text into the image. Meanwhile, the color image used is an image with a resolution of 512×512 pixels as many as fifteen pieces. The use of resolutions 512×512 because these resolutions are resolutions commonly used in similar researches. The color image used is in the format of Tagged Image File Format (*.tiff). The use of *.tiff format is because tiff is the basic format of digital images before compression and other manipulations are carried out. *.tiff format is also a flexible format and has very high image quality. The method proposed in this experiment is as shown in figure 1 below.

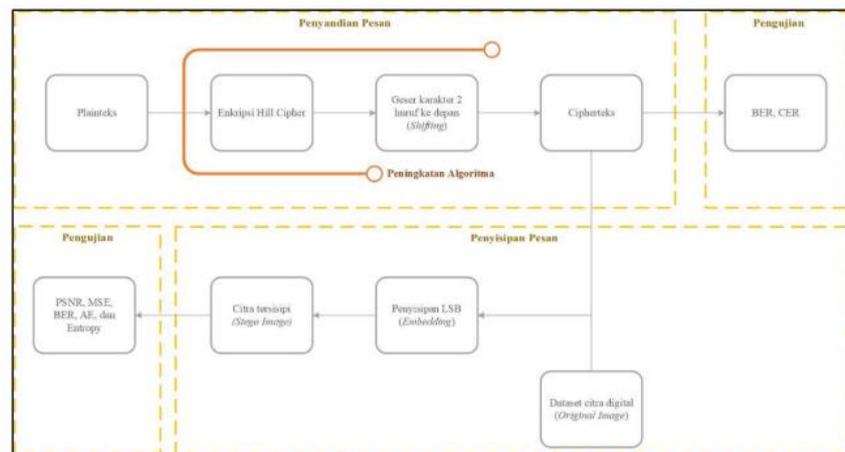


Figure 1. Proposed methods

In figure 1 above, there are three processes, namely the message encoding process, the message insertion process, and the message testing process. The whole process is explained as follows:

1. Enter or input text dataset (plaintext).
2. Once the text dataset is entered, then encrypt it along with the key with the Hill Cipher algorithm.
3. Next, before it is converted into ciphertext. Shift the character (Shifting) as much as 2 characters forward.
4. After shifting the characters, then convert it into a ciphertext.
5. Then, the ciphertext will be tested using the BER.28 testing tool
6. Next, if you have obtained ciphertext. Enter or input the digital image dataset as a container medium from ciphertext.
7. Next, do the insertion (embedding) using the LSB algorithm on the intensity of the red color channel.
8. After embedding, a Stego Image is obtained.
9. Finally, the inserted image (Stego Image) will be tested using MSE, PSNR, BER, AE, and Entropy testing tools.

Text messages will be encrypted using the Hill Cipher algorithm by converting the message into an ASCII model form and dividing the message into blocks according to the key matrix of each block. The result of the encryption will be shifted by two characters forward (shifting). It will then be converted into ciphertext. The overall process of encoding messages is described as follows:

1. Enter or input plaintext as text message and key matrix.
2. Convert the plaintext to decimal form as per ASCII table.

3. Divide the plaintext into blocks according to the key matrix.
4. Encode with the Hill Cipher algorithm.²⁹
5. Then, shift the character by 2 characters forward.
6. Finally, change back to the character form according to the ASCII table to get the ciphertext.

The encrypted message is then inserted into the original image using the LSB algorithm. The text message will be inserted into the red intensity Channel. Before insertion, text messages and digital imagery are first converted to binary form. The overall proposed message insertion model is as follows:

1. Input the digital image as a container medium from ciphertext.
2. Separate the digital image according to its intensity, then convert it to decimal form.
3. Change the ciphertext to binary form and convert the decimal conversion result of the digital image to binary form as well.³⁰
4. Insert ciphertext into a digital image with red channel intensity using LSB algorithm.
5. Convert the digital image binary back to decimal form, then combine each intensity into one to form a stego image.

Messages that have been hidden, will be revealed back to the original message. The proposed message return model can be described as follows:

1. Insert a stego image.
2. Divide the inserted imagery according to its colour intensity.³⁷
3. Perform extraction with LSB algorithm.
4. Then, a ciphertext will be obtained.
5. Change the ciphertext to decimal number form as per ASCII table.
6. Shift the character by 2 characters backwards (shifting).
7. Decrypt with the Hill Cipher algorithm along with the key.
8. Change the decimal number back to the character form according to the ASCII table.
9. Finally, you will get the plaintext.

In knowing the proposed method whether it succeeded in keeping text messages into a digital image well or not, it is necessary to evaluate by testing the quality of the stego image and comparing it with the original image. This stage is the stage where the stego image will be tested using several measurement tools.

Measurement tools that are usually used to determine good image quality or not by using:

Mean Square Error (MSE)

$$MSE = (1/M \times N) \sum_{(i=1..M)} \sum_{(j=1..N)} [(I_{ori}(i,j) - I_{dist}(i,j))]^2 \quad (1)$$

$M \times N$: ordo of digital image matrix

$I_{ori}(i,j)$: pixel value at point (i,j) in the original image

$I_{dist}(i,j)$: pixel value at point (i,j) in distorted image

Peak Signal to Noise Ratio (PSNR)

$$PSNR = 10 \log_{10} (MAX^2 / MSE) \quad (2)$$

MAX : the maximum pixel value in the image.

MSE : the MSE value calculated by equation (1).

Bit Error Rate (BER)

$$BER = |Bc - Bfc| \quad (3)$$

Bc : number of bits after the change

Bfc : number of bits before change

Avalanche Effect (AE)

$$AE = (bpb / jsb) \times 100\% \quad (4)$$

Bpb : Large bit change
 Jsb : Total number of bits

Meanwhile, the measurement tool used to measure the quality of encryption results is by using:

Bit Error Rate (BER) equation (3) and Character Error Rate (CER)

$$CER = \frac{jke}{pp} \quad (5)$$

jke : Error character count
 pp : Message length

Entropy.

$$e = -\sum p(f(x,y)) \cdot \log_2(p(x,y)) \quad (6)$$

$p(x,y)$: The pixel value to (x,y) .

$p(f(x,y))$: The ratio of many pixels $(f(x,y))$ to many pixels from the image.

From the tests obtained, the smaller the MSE value obtained, the better the quality of the stego image. Meanwhile, in PSNR, the greater the value of PSNR obtained, the better the quality of the stego image and similar to the original image. In BER encryption, the greater the BER value obtained, the more secure the quality of encryption. While in BER insertion, the smaller the BER value obtained, the better the quality of the stego image. Then the quality of stego image is getting better. Meanwhile, in PSNR, the greater the PSNR value obtained, the better the stego image quality and similar to the original image. In BER encryption, the greater the BER value obtained, the more secure the quality of encryption. While in BER insertion, the smaller the BER value obtained, the better the quality of the stego image. The AE value is close to 1 or the percentage is close to 100%, so the algorithm used is effective. Just like with BER, AE can be used in cryptography and steganography. CER has a job of calculating how much error a character makes after it is rerevealed. CER can be used in cryptography and steganography. The method can be said to be more perfect if it has a CER value close to 0. The result of the Entropy calculation is used to show the measure of the irregularity of the shape of an image. If the resulting entropy value is large, then the structure of the image is orderly. Conversely, if the resulting entropy value is small then the image structure is irregular.

2.1. Equations

Number equations consecutively with equation numbers in parentheses flush with the right margin, as in (1). To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use an en dash (–) rather than a hyphen for a minus sign. Use parentheses to avoid ambiguities in denominators. Punctuate equations with commas or periods when they are part of a sentence, as in

3. RESULTS AND DISCUSSION

In this study the message used to insert into the picture was a text message. Text messages used are 32, 64, 128 and 14,628 characters and the characters used are only letters of the alphabet from 'a' to 'z' without spaces, numerics, symbols and other special characters. Because, the character used to divide the value of a character is only 26. The use of such character length is due to prove the insertion of text into the image. In this study the message used to insert into the picture was a text message. Text messages used are 32, 64, 128 and 14,628 characters and the characters used are only letters of the alphabet from 'a' to 'z' without spaces, numerics, symbols and other special characters. Because, the character used

to divide the value of a character is only 26. The use of such character length is due to prove the insertion of text into the image.

The key used in this study is a matrix. The matrix used is a matrix of order 2×2 . The matrix must be invertible and its determinant value must be prime relative to 26. In 60 experiments will use the same matrix. The values of the matrix are as follows:

$$\begin{pmatrix} 55 & 8 \\ 17 & 3 \end{pmatrix}$$

The proposed method is divided into 3 (three) stages, namely the message encoding process (encryption), the message hiding process (embedding) and the testing process. In the message encoding section, an analysis of the message encoding process (encryption) was carried out using the Hill Cipher algorithm. The message will be converted in an ASCII number. Then the ASCII number is reduced by 97. The result after subtracting 97, is further divided into certain blocks according to the order of the key. The keys used in this study have orders of order 2×2 meaning each block will hold 2 characters. So if the message consists of 32 characters, 16 blocks will be formed. The results in each block are then applied to encryption calculations, such as the calculation examples in blocks 1 and 2 as follows:

Block 1 : 20 and 13 results from 2 characters:

$$"u" = 117 \text{ (ASCII)} - 97 = 20$$

$$"n" = 110 \text{ (ASCII)} - 97 = 13$$

$$\text{block 1} = \begin{vmatrix} 55 & 8 \\ 17 & 3 \end{vmatrix} \begin{vmatrix} 20 \\ 13 \end{vmatrix} \text{ mod } 26 = \begin{vmatrix} 8 \\ 15 \end{vmatrix}$$

Block 2 : 8 and 21 results of 2 characters:

$$"i" = 105 \text{ (ASCII)} - 97 = 8$$

$$"v" = 118 \text{ (ASCII)} - 97 = 21$$

$$\text{block 1} = \begin{vmatrix} 55 & 8 \\ 17 & 3 \end{vmatrix} \begin{vmatrix} 8 \\ 21 \end{vmatrix} \text{ mod } 26 = \begin{vmatrix} 10 \\ 17 \end{vmatrix}$$

From the calculation above, ciphertext numbers are obtained. before shifting and converting to character form. Then shift 2 (two) characters forward before being converted to the form of characters according to the ASCII table, so that the values of block 1 (8 and 15) will move to block 2, the values of block 2 (10 and 17) will move to block 3, and so on. Fill in the last block values (9 and 5) will be the values in block 1.

After shifting 2 characters forward, the last step is to convert the numbers into the form of a character by adding the numbers as many as 97. These additions are used so that they can be converted according to ASCII tables. The conversion of ciphertext numbers to the form of the cipher characters is as follows:

$$1\text{st block. } 9 + 97 = 106 \rightarrow \text{j character}$$

$$5 + 97 = 102 \rightarrow \text{f character}$$

$$2\text{nd block. } 8 + 97 = 105 \rightarrow \text{i character}$$

$$15 + 97 = 112 \rightarrow \text{p character}$$

$$3\text{rd block. } 10 + 97 = 107 \rightarrow \text{k character}$$

$$17 + 97 = 114 \rightarrow \text{r character}$$

The next process is the process of hiding secret messages that have been encrypted previously into digital image media. This concealment or insertion process uses the Least Significant Bit (LSB) steganography method. The concealment or sweeping process will be

carried out at 1 (one) color intensity only, namely at the red intensity. The first step in the concealment process is to take a digital image that will be used as a concealment medium. The digital image used in this study is a Red, Green, Blue (RGB) color image with a resolution or dimension of 512 × 512 pixels which can be seen in figure 2. In this case the author will use the image "Pappers.tiff" as an example of processing.



Figure 2. Image "pappers.tiff"

The image will be converted to the form of an integer number and divide it into each color intensity. After converting the image to the form of integer numbers as above, then convert the secret message that has been encrypted before, into binary form. For example 5 characters in blocks 1, 2, and 3 (j, f, i, p, and k) as in table 1.

Table 1. Convert Character to Binary

character	ASCII Number	Binary							
j	106	0	1	1	0	1	0	1	0
f	102	0	1	1	0	0	1	1	0
i	105	0	1	1	0	1	0	0	1
p	112	0	1	1	1	0	0	0	0
k	107	0	1	1	0	1	0	1	1

Changing the secret message in the table above for each character will result in 8 bits. Of those 8 bits multiplied by the number of characters (5 characters) that results in 40 bits. Because it produces 40 bits, at the intensity of the red color channel a digital image will be taken the first 40 integer numbers out of 262,144 integer numbers. The integer number will be converted to binary form, then on the last bit each integer number will be compared sequentially with the bits of the secret message, if the last bit is the same value as the bit of the secret message, then there is no change in value or fixed value. Whereas if the last bit is greater or smaller than the secret message bit, then there is a change in value. Because in the study the insertion was only carried out at the intensity of the red color. So that at the intensity of green and blue there is no need to change bits.

From the process of hiding the message, the original image and the inserted image (stego image) at first glance in the human sense vision both images do not change or look similar. But both have differences or experience changes although not significantly. These changes can be seen through testing using Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR).

The results of encoding with Hill Cipher can be known that if a text message (plaintext) that has the same characters does not necessarily produce the same encryption result (ciphertext) as well. However, the character length will still result in the same character length. If the result of encryption (ciphertext) will change the value of a character from the original message (plaintext). So, later it will affect the insertion process with LSB. From the evaluation results using Bit Error Rate (BER) it can be concluded that, if the longer the message used, the

greater the BER value obtained or the more bit errors. While from the evaluation using CER there are no character errors when ciphertext is decrypted, this shows this algorithm is very good. But in terms of encryption time, text messages with a character length of 64 characters have a fast travel time of 0.00117655 seconds which is faster than 32 characters, while with a length of 128 and 14,628 characters have a slow travel time of 0.00161681 and 0.0414762 seconds.

When observed from the results of insertion with LSB, it can be seen that all images from insertion are similar to the original image. This proves LSB is one of the good methods in inserting a message. However, in terms of insertion travel time, the longer the text message to be inserted, the more time it takes. To find out which inserted image is the best, it is necessary to observe with MSE and PSNR.

The results of the MSE evaluation of insertion in 15 images show that, the longer the inserted message, the greater the MSE value to be obtained. This can be seen from the results of MSE calculations in figures 3, 4, 5 and 6.



Figure 3. SME result with insertion of 32 characters



Figure 4. SME result with 64-character insertion



Figure 5. SME result with 128-character insertion



Figure 6. SME result with insertion of 14,628 characters

From some of these images, it can also be seen that each image will produce a different MSE because it is influenced by the integer value of an image. At insertions of 32 and 64 characters, Mandrill_Baboon.tiff image is the image with the lowest MSE value having an MSE value of 0.0001450 for 32 characters and 0.0002886 for 64 characters. Meanwhile, at the insertion of 128 characters, the image San_Diego_Shelter_Island.tiff into the image with the lowest MSE154 value which has an MSE value of 0.0007184. For insertions of 14,628 characters, the San_Diego_North_Island_NAS.tiff image is the image with the lowest MSE of 3.329312633. In the previous chapter it has been explained, the smaller the MSE value, the better the quality of the resulting image. From the evaluation results of PSNR table 4.21 above, it can be seen that inserting messages with a length of 32 characters results in an average PSNR above 85 db, while inserting messages with a length of 64 characters will decrease by 2-3 db. For insertion of messages with a length of 128 characters will result in an average PSNR of 74 db. However, if inserted with a large character length then the PSNR will fall below 50 db. From the table, it can also be seen that the image that has the highest PSNR value is the Mandrill_Baboon.tiff image with a 32-character message insert that has an average PSNR value of 86.5184. While the image that has the lowest PSNR value is the San_Francisco_Golden_Gate.tiff image with a 128-character message insert that has an average PSNR value of 78.7751. For inserts of 14,628 characters of NAS San_Diego_North_Island_images.tiff is the image with the highest PSNR, with an average PSNR of 42,930. However, the blue and green PSNR values will produce infinite values, because insertion is only done at color intensity. Thus, the intensity of blue and green remains the same as the original image. So, it can be concluded that the more characters inserted and the greater the MSE value obtained by an image, the smaller the PSNR value that will be obtained and the worse the image quality.

In the results of the insertion BER evaluation, it can be shown that, BER with an insertion of 32 characters has an average value below 45 of the 256 bits inserted. This means that 17.5% of bits change from the 256 bits inserted. Meanwhile, the BER value with the insertion of 64 characters obtained an average value below 90 of the 512 bits inserted. So there are about 17.5% of bits that change from the 512 bits inserted. For BER values with an insertion of 128 characters will result in an average value of under 180 out of 1024 bits. It has 17.5% bit change from 1024 bits. Meanwhile, from the insertion of 14,628 characters will 155 result in a BER below 36,200 of 117,024 bits. There are about 30% of the 117,024 bit turns. The smaller the BER value obtained and the larger the insertion bit, the smaller the occurrence of bit errors. It also relies on changing the pixel bit of the image with the message bit. The smaller the BER value obtained, the more image quality is similar to the original image. The BER value will also affect the Avalanche Effect (AE) value.

The evaluation results with Avalanche Effect (AE), this resulted in an AE of 0.05% at the insertion of 32 characters. Meanwhile, at the insertion of 64 AE characters, the red intensity produces a value of 0.09%. For insertion of 128 characters results in a red intensity AE of 0.19% and insertion of 14,628 characters results in a red intensity AE of 41.4%. So it can be concluded that the larger the inserted character, the greater the AE value obtained. The greater the AE value obtained, the more resistant it is to the attack of manipulation operations carried out on the stego image. The evaluation results use Entropy as shown in figures 7, 8, 9 and 10.

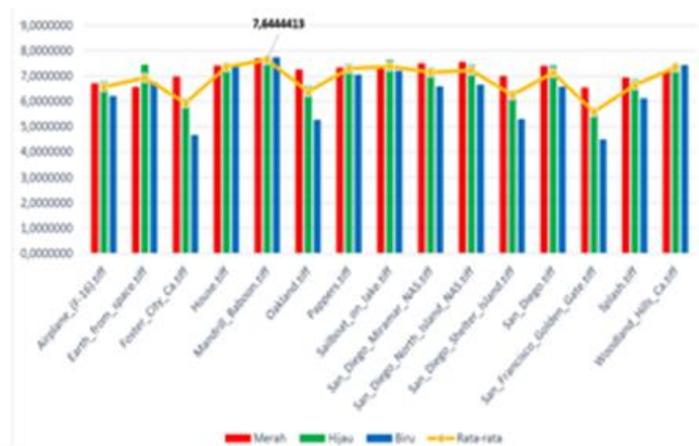


Figure 7. Entropy result with 32 characters Insertion

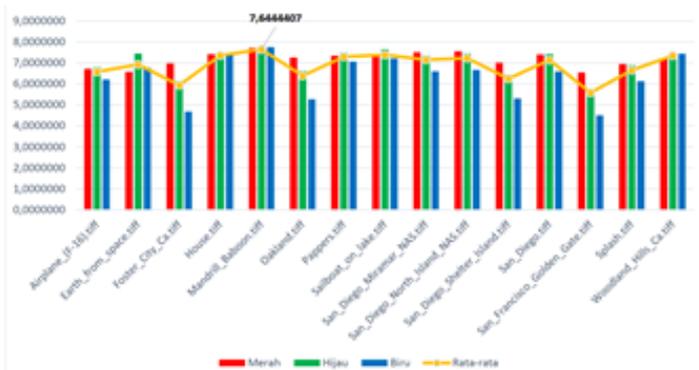


Figure 8. Entropy results with 64-character insertion

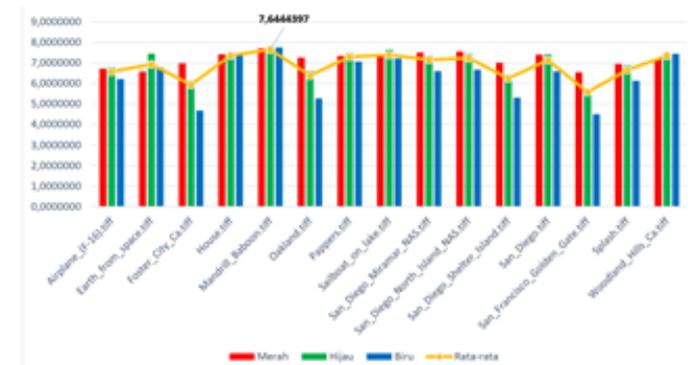


Figure 9. Entropy results with 128-character insertion

The results of this experiment can be developed for future research using a combination of diverse characters, Message masking can be improved not only on digital images but using other media such as video and audio. Further development also required testing of attacks such as brightness, contrast, saturation, or on exposure.

REFERENCES

- [1] Sari, J. I., Dan Sihotang, H. T., Implementasi Penyembunyian Pesan Pada Citra Digital Dengan Menggabungkan Algoritma HILL Cipher Dan Metode Least Significant BIT (LSB), *Jurnal Mantik Penusa*, 1(2).J. Breckling, Ed., *The Analysis of Directional Time Series: Applications to Wind Speed and Direction*, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61. 2017.
- [2] Nofriansyah, D., Defit, S., Nurcahyo, G. W., Ganefri, G., Ridwan, R., Ahmar, A. S., and Rahim, R., A New Image Encryption Technique Combining Hill Cipher Method, Morse Code and Least Significant Bit Algorithm. In *Journal of Physics: Conference Series* (Vol. 954, p. 012003). IOP Publishing. 2018.
- [3] Sari, C. A., Rachmawanto, E. H., Utomo, D. W., dan Sani, R. R., Penyembunyian Data Untuk Seluruh Ekstensi File Menggunakan Kriptografi Vernam Cipher dan Bit Shiffting, *Journal of Applied Intelligent System*, 1(3), 179-190. 2016.
- [4] Pradipta, A., Implementasi Metode Caesar Chiper Alphabet Majemuk Dalam Kriptografi Untuk Pengamanan Informasi. *Indones, J. Netw. Secur*, 5(3), 3-6, 2016.
- [5] Basri, B., Kriptografi Simetris dan Asimetris dalam Perspektif Keamanan Data dan Kompleksitas Komputasi, *Jurnal Ilmiah Ilmu Komputer*, 2(2)., 2016.
- [6] Kusuma, E. J., Sari, C. A., Rachmawanto, E. H., and Moses Setiadi, D. R. I., A Combination of Inverted LSB, RSA, and Arnold Transformation to get Secure and Imperceptible Image Steganography, *Journal Of ICT Research & Applications*, 12(2), 2018.
- [7] Kaur, A., A Review on Symmetric Key Cryptography Algorithms, *International Journal of Advanced Research in Computer Science*, 8(4), 2017.
- [8] Handoko, L. B., Umam, C., and Anindita, A. S., Upload file security on the server using LSB and Hill Cipher, *Journal of Applied Intelligent System*, 4(1), 28-38, 2019.
- [9] Mezher, Liqaa Saadi, and Ayam Mohsen Abbass. Mixed Hill Cipher methods with triple pass protocol methods. *International Journal of Electrical and Computer Engineering* 11.5 (2021): 4449.
- [10] Mfungo, Dani Elias, et al. Enhancing Image Encryption with the Kronecker xor Product, the Hill Cipher, and the Sigmoid Logistic Map. *Applied Sciences* 13.6 (2023): 4034
- [11] Gunawan, I., Sumarno, S., Irawan, E., dan Tambunan, H. S., Pengamanan Berkas Dokumen Menggunakan Fungsi Algoritma Steganografi LSB. *ALGORITMA: Jurnal Ilmu Komputer dan Informatika*, 2(1), 2018.
- [12] Jatmoko, C., Handoko, L. B., dan Sari, C. A., Uji Performa Penyisipan Pesan Dengan Metode LSB dan MSB Pada Citra Digital Untuk Keamanan Komunikasi. *Dinamika Rekayasa*, 14(1), 47-56, 2018.
- [13] Adam, Riza Ibnu. Skema Penyembunyian Data pada Gambar Berbasis Interpolasi Kubik B-Spline Menggunakan Metode Least Significant Bit (LSB), *JEPIN (Jurnal Edukasi dan Penelitian Informatika)* 5.3, 255-260, 2019.
- [14] Harahap, Muhammad Khoiruddin, dan Nurul Khairina. Steganography Citra dengan Metode Least Significant Bit Random Placement, *JEPIN (Jurnal Edukasi dan Penelitian Informatika)* 6.2: 245-249, 2020.

- [15] Megantara, R. A., Rafrastara, F. A., and Mahendra, S. N., A combination of Hill CIPHER-LSB in RGB image encryption, *Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control*, 241-248, 2019.
- [16] Abdillah, Muhammad Oemar, Ogie Ariansah Pane, and Farhan Rusdy Asyhary Lubis. Implementasi Keamanan Aset Informasi Steganografi Menggunakan Metode Least Significant Bit (LSB), *Jurnal Sains dan Teknologi (JSIT)* 3.1, 40-46, 2023.
- [17] Pandey, A., Pandey, S., and Agarwal, A. K., Transpoly Hill Cipher--An Improvement Over Traditional Hill Cipher, *International Journal of Advanced Research in Computer Science*, 9(1), 2018.